SENA

PROGRAMA DE FORMACIÓN

ANÁLISIS EXPLORATORIO DE DATOS EN PYTHON

INSTRUCTOR

LUZ NEIRA VARON PEÑA

INFORME DE RESULTADOS OBTENIDOS DEL ANÁLISIS DE DATOS EXPLORATORIO REALIZADO AL CASO DE ESTUDIO. AA4-EV01.

APRENDIZ

ENMANUEL A. DUARTE CÁCERES

COLOMBIA

2025

Introducción

En el presente informe se presenta la evidencia de implementación de las herramientas y librerías necesarias para el análisis de datos en Python para el caso de estudio correspondiente a la semana 4.

A lo largo del documento se detallan los pasos para la correcta manipulación, lectura, ordenamiento y limpieza de datos, correspondiente al manejo de la infraestructura para el correcto análisis de datos, así mismo se realizan diferentes cálculos estadísticos y se visualizan para la correcta interpretación.

Finalmente, el documento muestra las conclusiones de la actividad.

Caso de estudio

El dataset utilizado para el desarrollo de la actividad se encuentra en el archivo CSV titulado "Data_Caso_Propuesto" el cual por temas de lectura se renombro "inmuebles". A continuación, se muestra una tabla correspondiente al tipo de variable y las columnas que lo conforman.

Tabla 1. Variables del Dataset.

Variables Categóricas	Variables Numéricas
Ciudad (object)	Codigo (int64)
Departamento (object)	Area Terreno (float64)
Barrio (object)	Area Construida (float64)
Direccion (object)	Precio (float64)
Detalle disponibilidad (object)	
Estrato (object)	
Tipo Inmueble (object)	
Datos adicionales (object)	

Preguntas objetivo:

1. ¿Cuál es la relación entre el estrato y el precio de los inmuebles disponibles para la venta, según el tipo de inmueble?
2. ¿Qué departamentos concentran la mayor cantidad y el mayor valor total de inmuebles disponibles para la venta?
3. ¿Qué diferencias existen en los precios promedio de los inmuebles según el estrato socioeconómico declarado?

Para la correcta manipulación debemos hacer uso de diferentes librerías, las cuales nos ayudaran a reducir el tiempo de ejecución y nos brindan las herramientas necesarias para el correcto procesamiento y análisis de datos.

Tabla 2. Librerías utilizadas.

Librería	Descripción
Pandas	Usada para la manipulación y visualización de grandes volúmenes de datos Comando: `Import pandas as pd`
Matplotlib	Permite generar muy fácilmente diversos tipos de gráficos. Comando: `Import matplotlib.pyplot as plt`

Seaborn	Permite la gestión de gráficos más atractivos de Matplotlib, gráficos informativos y estadísticos. Comando: `Import seaborn as sns`

Creación del Dataframe

Para iniciar debemos crear un dataframe que lea los datos en nuestro CSV

```
import pandas as pd
df = pd.read_csv('inmuebles.csv')
```

y visualizamos la información en el dataframe

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 463 entries, 0 to 462
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Codigo                  463 non-null    int64
 1   Ciudad                  463 non-null    object
 2   Departamento            463 non-null    object
 3   Barrio                  40 non-null     object
 4   Direccion               463 non-null    object
 5   Area Terreno            463 non-null    float64
 6   Area Construida         463 non-null    float64
 7   Detalle Disponibilidad  463 non-null    object
 8   Estrato                 463 non-null    object
 9   Precio                  463 non-null    float64
 10  Tipo de Inmueble        463 non-null    object
 11  Datos Adicionales       118 non-null    object
```

Se puede visualizar en la información de los datos que el resultado para las columnas Barrio y Datos Adicionales son de 40 y 118, respectivamente, mientras que para las demás columnas el valor es 463.

Barrio: Solo tiene 40 valores de 463 lo que quiere decir que más del 90% está vacío. Es difícil imputar o completar correctamente sin información externa (como geolocalización).

Datos Adicionales: Solo 118 valores completos (≈ 25%) y probablemente con información no estructurada (texto libre).

Por lo que bajo el criterio, eliminamos ambas columnas

```
df = df.drop(['Barrio', 'Datos Adicionales'],axis=1)
```

y con el siguiente comando eliminamos los duplicados

```
df = df.drop_duplicates()
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 463 entries, 0 to 462
Data columns (total 10 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Codigo                  463 non-null    int64
 1   Ciudad                  463 non-null    object
 2   Departamento            463 non-null    object
 3   Direccion               463 non-null    object
 4   Area Terreno            463 non-null    float64
 5   Area Construida         463 non-null    float64
 6   Detalle Disponibilidad  463 non-null    object
 7   Estrato                 463 non-null    object
 8   Precio                  463 non-null    float64
 9   Tipo de Inmueble        463 non-null    object
```

Ahora visualizamos la información en la siguiente tabla:

Tabla 3. Tabulación de los datos.

Total, registros	463
Registros sin valores nulos	463
Columnas eliminadas	2
Total, nulos	0
Total, duplicados	0
Registro sin nulos y duplicados	463

Análisis estadístico

A continuación, se utiliza la función describe(), el cuál permite calcular las medidas de tendencia central y dispersión, aplicando algunos métodos estadísticos como la media, mediana, desviación estándar y cuartiles.

```
df.describe()
```

Tabla 4. Medidas de tendencia central y dispersión.

	Codigo	Area Terreno	Area Construida	Precio
count	463.000000	4.630000e+02	463.000000	4.630000e+02
mean	18003.151188	1.515204e+04	87.517279	6.672032e+08
std	1992.191499	1.827101e+05	1137.469077	3.272992e+09
min	2575.000000	0.000000e+00	0.000000	4.650000e+06
25%	18184.500000	0.000000e+00	0.000000	1.230500e+07
50%	18332.000000	0.000000e+00	0.000000	1.587000e+07
75%	18539.500000	0.000000e+00	0.000000	1.379955e+08
max	19344.000000	3.217197e+06	22724.000000	4.523379e+10

De acuerdo con la información mostrada, el área de terreno y área construida no se debe tener en cuenta como valores importantes, ya que presentan valores de cero en su mayoría. Esto puede deberse a que no se tenían los valores en el registro y los ingresaron como 0, sin embargo, podemos analizar de acuerdo al precio y agruparlos por percentiles.

Ordenamiento y agrupación de datos

Definimos los rangos

```
rangos = [0, 1e7, 5e7, 1e8, 5e8, 1e9, float('inf')]
```

Y asignamos un nombre o etiqueta a cada uno

```
nombrerangos = ['<10M', '10M-50M', '50M-100M', '100M-500M',
'500M-1000M','>1000M']
```

Ahora con este nuevo agrupamiento podemos crear una nueva variable

```
df['Rango_Precio'] = pd.cut(df['Precio'], rangos, labels =
nombrerangos)
```

con el siguiente comando se puede visualizar los primeros registros del dataframe,

```
df.head()
```

Tabla 5. Visualización primeros registros.

Codigo	**Ciudad**	**Departamento**	**Direccion**	**Area Terreno**	**Area Construida**	**Detalle Disponibilidad**	**Estrato**	**Precio**	**Tipo de**	**Rango_Precio**

									Inmueble	
17180	BOGOTA	CUNDINAMARCA	AV CR 7 NO. 166 - 51 LT B	0.00	0.0	COMERCIALIZABLE CON RESTRICCION	TRES	2.958081e+10	LOTE COMERCIAL	>1000M
19292	BOGOTA	CUNDINAMARCA	CL 72 No. 12 - 77	0.00	0.0	COMERCIALIZABLE	COMERCIAL	1.646059e+10	EDIFICIO	>1000M
19292	BOGOTA	CUNDINAMARCA	CL 72 No. 12 - 77	0.00	0.0	COMERCIALIZABLE VENTA ANTICIPADA	COMERCIAL	1.646059e+10	EDIFICIO	>1000M
2575	SOGAMOSO	BOYACÁ	CRA. 10 #11-78/80 Ó CL 12 # 9 - 77/85 Ó CALLE...	1655.08	7269.0	COMERCIALIZABLE CON RESTRICCION	CUATRO	1.376828e+10	CLINICA	>1000M
11409	BUGA	VALLE DEL CAUCA	LT A1-A24 B1-B79 C1-C51 D1-D9 STA ROSA LT1-46 ...	3217197.00	22724.0	COMERCIALIZABLE FIDUCIA	RURAL	4.523379e+10	LOTE MIXTO	>1000M

Queremos saber cuántos registros tiene cada rango, y utilizamos el siguiente comando

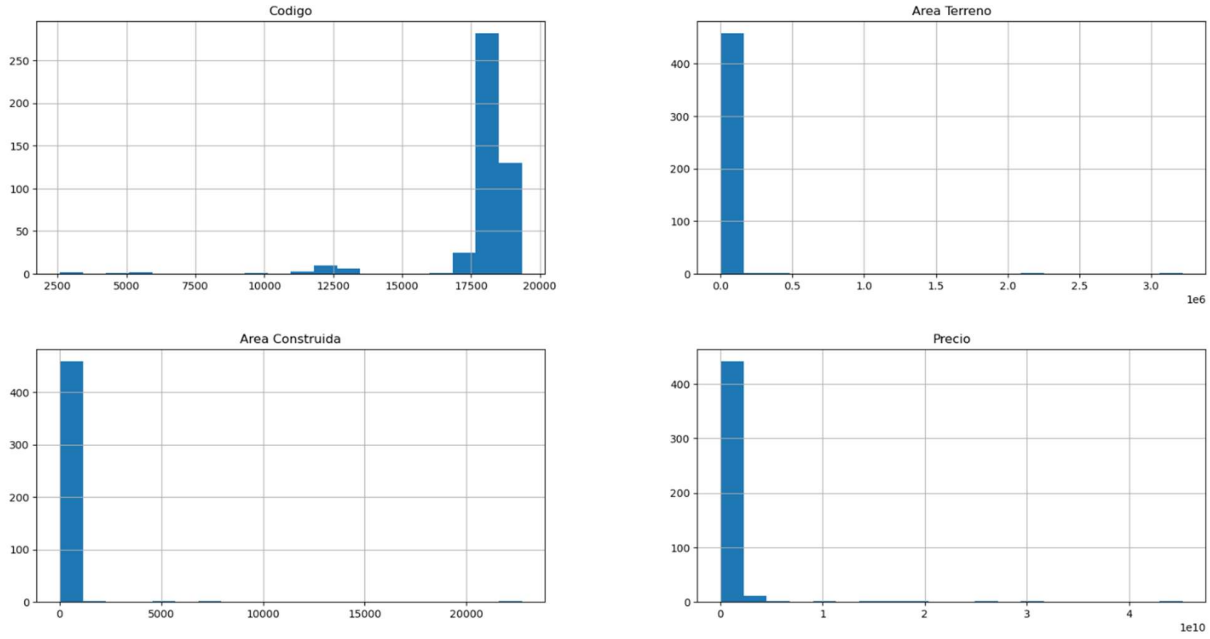
```
print(df['Rango_Precio'].value_counts().sort_index())
Rango_Precio
<10M          69
10M-50M      259
50M-100M      13
100M-500M     45
500M-1000M    28
>1000M        49
```

Gráficos

Para una mejor comprensión de los datos se utilizan los gráficos, en Python podemos utilizar histogramas de frecuencia, gráficos de barras, gráfico de torta, gráficos de cajas y bigote.

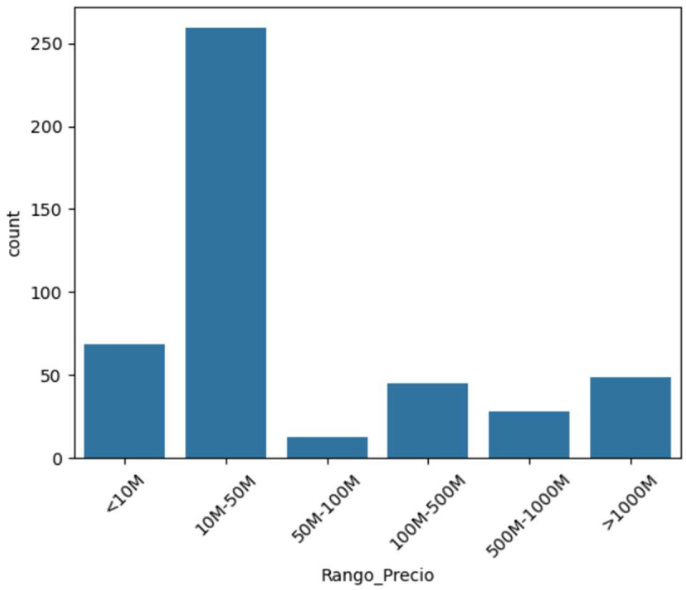
Histograma de frecuencia

```
df.hist(bins=20, figsize=(20,10))
```



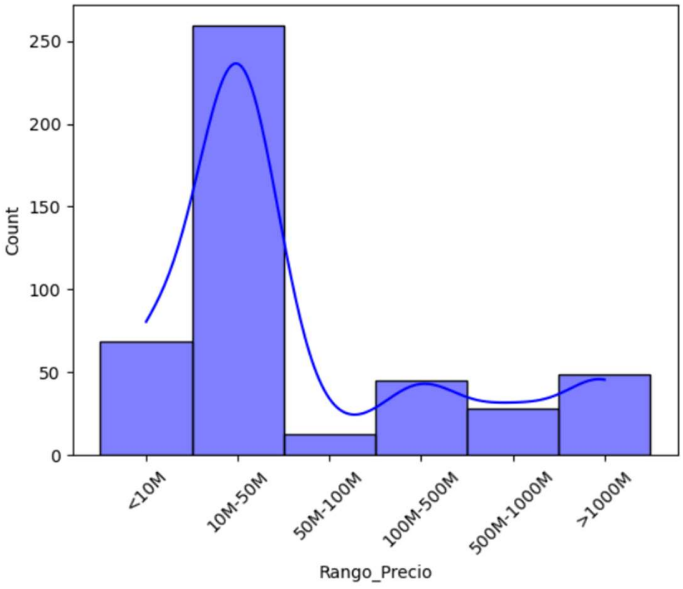
Las anteriores gráficas no nos son de mucha utilidad ya que la desviación estándar es demasiado grande, por lo que nos centraremos en el rango de precios,

```
plt. Figure(figsize=(10,7))
sns.countplot(x= df.Rango_Precio)
plt.xticks(rotation=45)
plt.show()
```



Para mejorar estéticamente los gráficos se utiliza el siguiente comando, con el cuál podremos visualizar una línea suavizada mostrando la distribución de los datos.

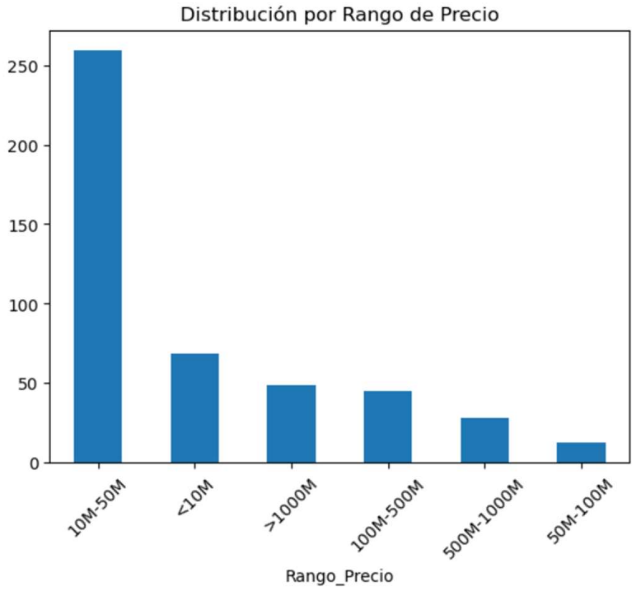
```
import seaborn as sns
import matplotlib.pyplot as plt
sns.histplot(df.Rango_Precio, color ="b", bins = 30, kde = True)
plt.xticks(rotation=45)
plt.show()
```



Con el siguiente comando podemos ver la información de manera descendente

```
df['Rango_Precio'].value_counts().plot(kind='bar',
title='Distribución por Rango de Precio')
```

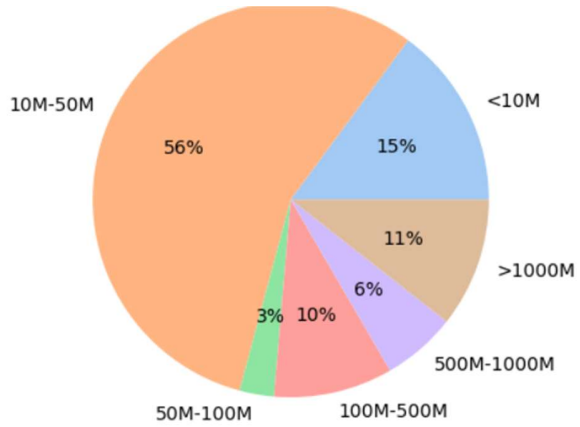
```
plt.xticks(rotation=45)
plt.show()
```



A partir de los histogramas de frecuencia y grafico de barras podemos decir que hay una mayor agrupación de registros en el Rango de 10 a 50 M, y el rango de 50 a 100 M tiene la menor cantidad de registros.

Gráfico de Torta

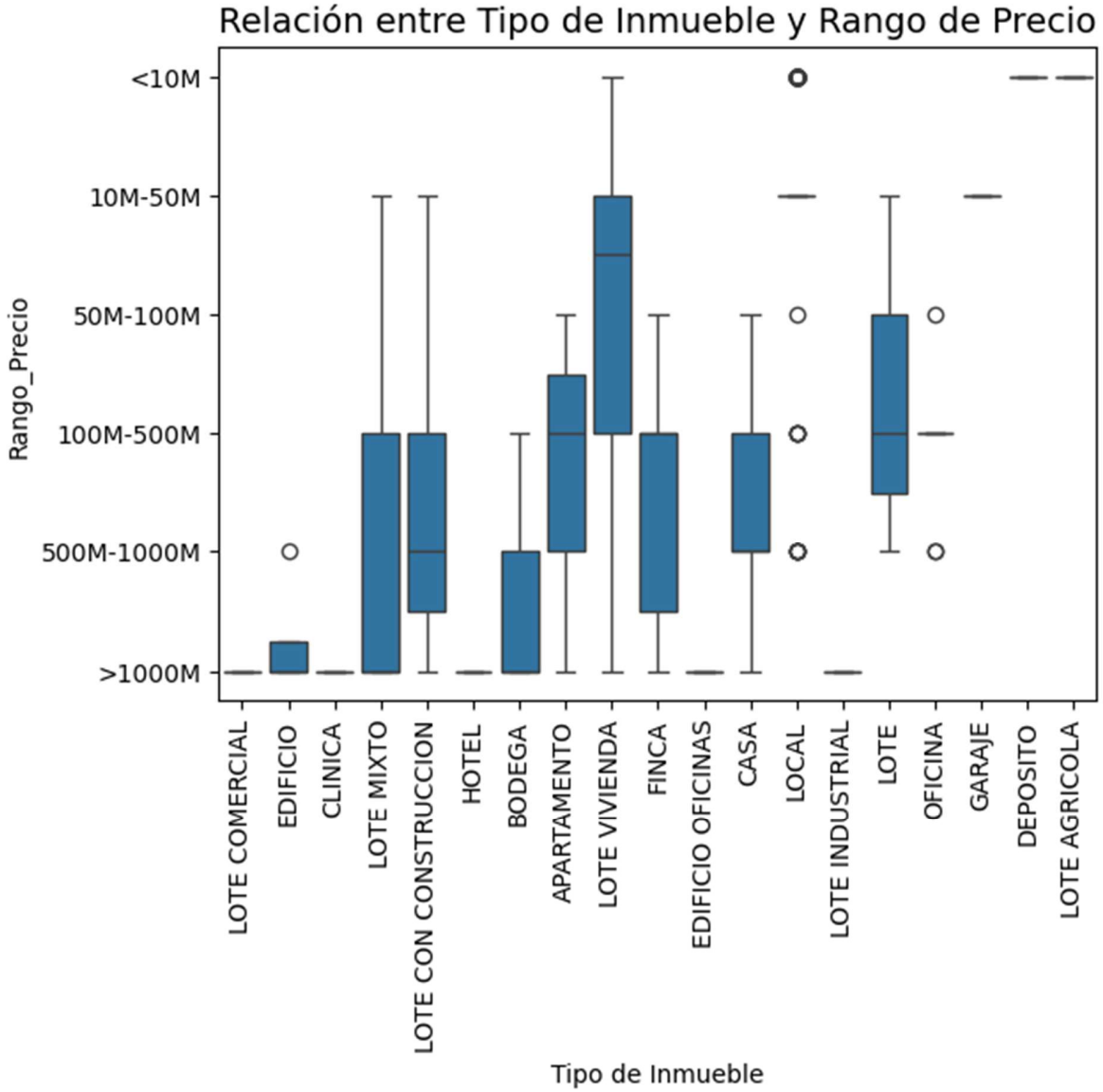
```
total_precio = df['Precio'].groupby(df['Rango_Precio'], observed
= True).count()
etiquetas = total_precio.index
colors = sns.color_palette('pastel')[0:6]
plt.pie(total_precio, labels = etiquetas, colors = colors,
autopct='%.0f%%')
plt.show()
```



De manera porcentual se puede establecer que alrededor del 70% de los registros están en el rango comprendido hasta los 50 M, y sorprende ver que el 11% supera el limite del rango de 1000 M

1. ¿Cuál es la relación entre el estrato y el precio de los inmuebles disponibles para la venta, según el tipo de inmueble?

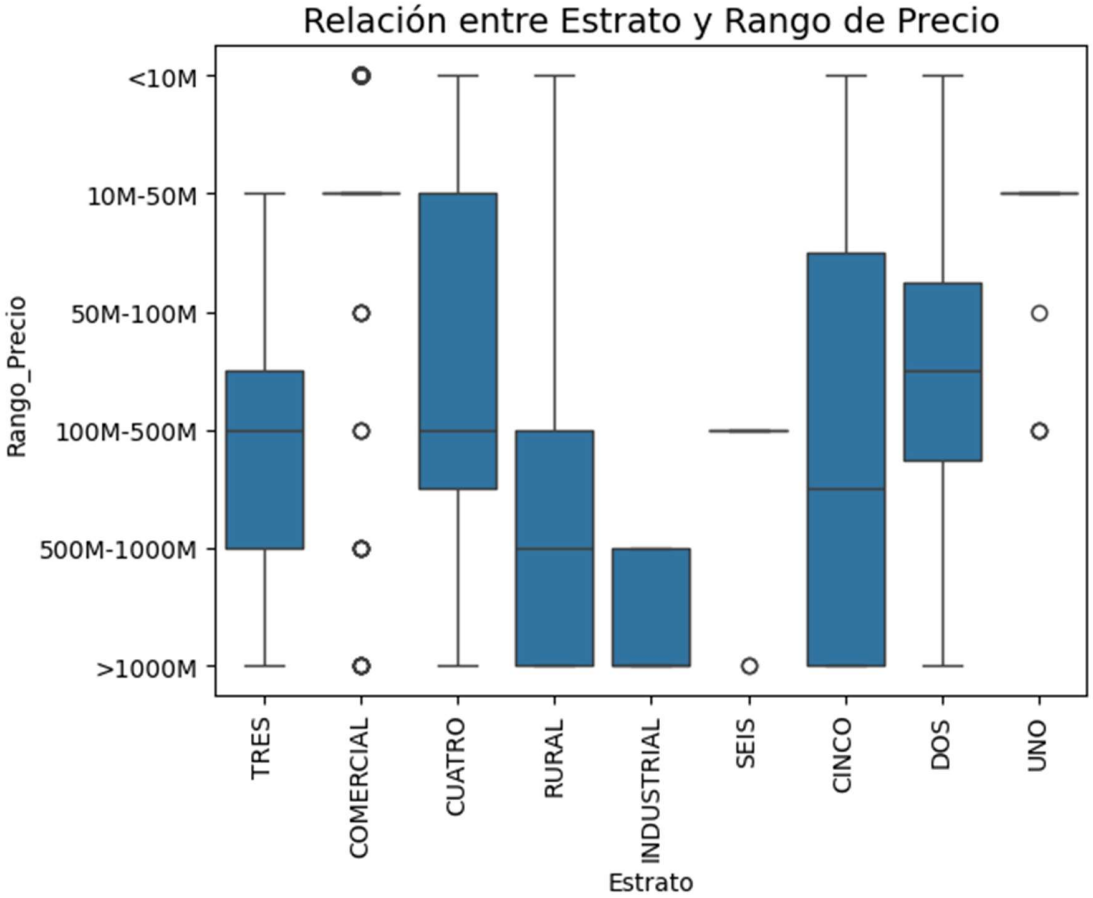
```
inmueble_valor = sns.boxplot(x=df["Tipo de Inmueble"],
y=df["Rango_Precio"])
plt.xticks(rotation=90, ha='center')
plt.title("Relación entre Tipo de Inmueble y Rango de Precio",
fontsize=14)
plt.show()
```



Como era de esperarse los precios mas elevados corresponden a lotes comerciales, bodegas, hoteles, lotes industriales, edificios de oficina y clínicas, esto se debe a la

naturaleza del sector retail, ya que están directamente relacionados a los servicios y productos que mueven la economía.

```
estrato_valor = sns.boxplot(x=df["Estrato"],
y=df["Rango_Precio"])
plt.xticks(rotation=90, ha='center')
plt.title("Relación entre Estrato y Rango de Precio",
fontsize=14)
plt.show()
```

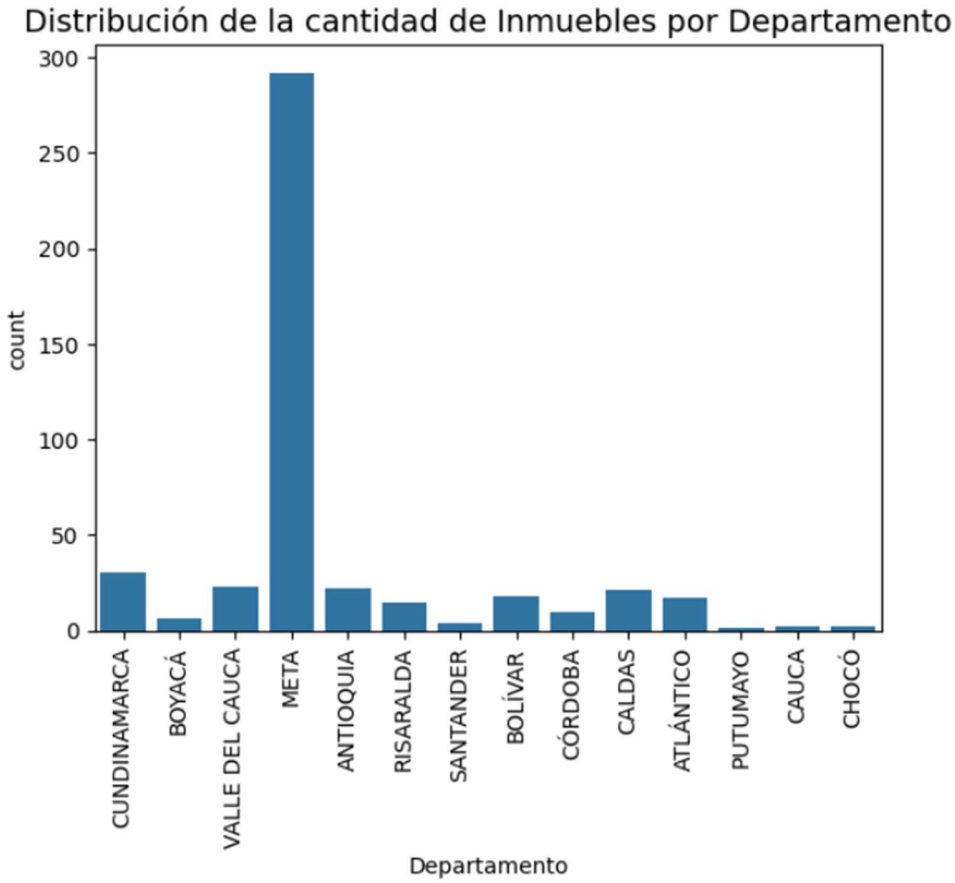


A partir del gráfico podemos concluir que las medianas con un mayor precio corresponden a los estratos rural, industrial y cinco, aunque hay algunos valores atípicos en el estrato comercial y seis lo que indica que hay propiedades fuera de los rangos normales.

2.¿Qué departamentos concentran la mayor cantidad y el mayor valor total de inmuebles disponibles para la venta?

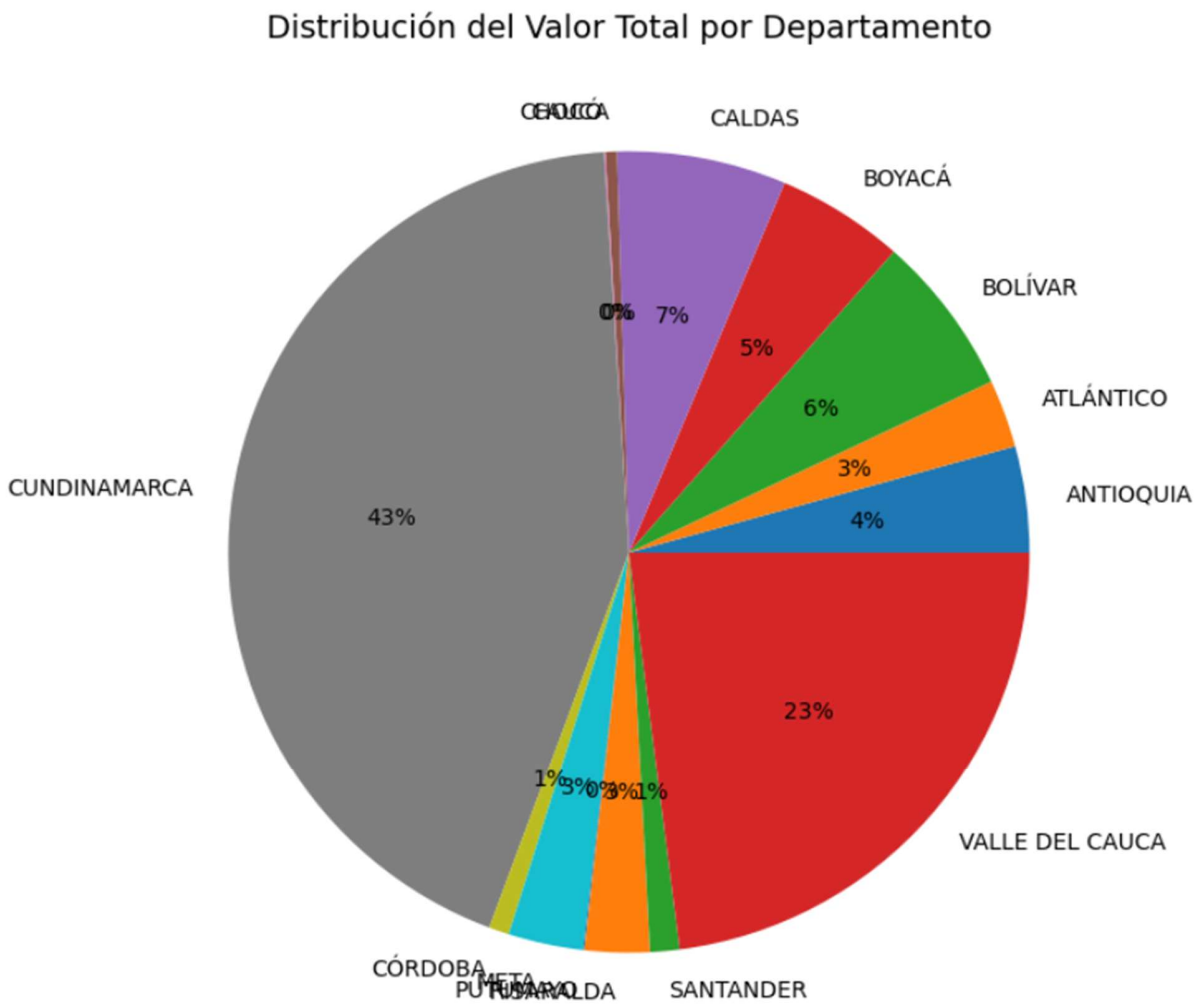
```
plt. Figure(figsize=(10,7))
sns.countplot(x= df.Departamento)
```

```
plt.title('Distribución de la cantidad de Inmuebles por
Departamento', fontsize=14)
plt.xticks(rotation=90, ha='center')
plt.show()
```



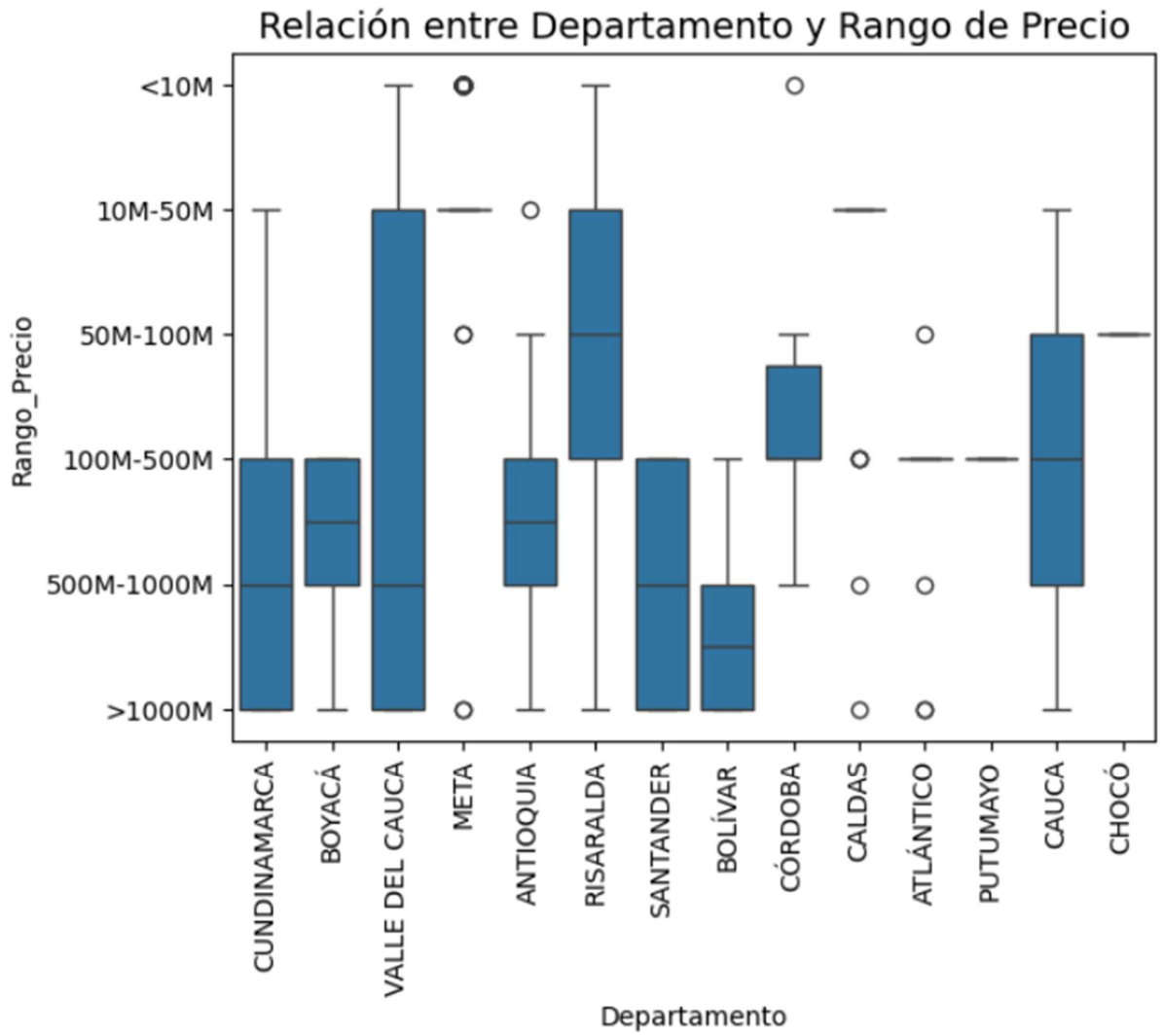
A partir del histograma de frecuencia notamos que el departamento con mayor numero de inmuebles es el departamento del Meta, con más de 250 inmuebles, es por mucho el que mayor concentración tiene, y por otra parte el Choco, Cauca, Putumayo y Santander, concentran la menor cantidad de inmuebles.

```
Valor_total_por_departamento =
df.groupby('Departamento')['Precio'].sum()
plt.figure(figsize=(10, 8))
plt.pie(valor_total_por_departamento.values, labels =
valor_total_por_departamento.index, autopct
='%.0f%%',textprops={'fontsize': 10})
plt.title('Distribución del Valor Total por Departamento',
fontsize=14)
plt.show()
```



Anteriormente vimos como el Meta concentraba la mayor cantidad de inmuebles, sin embargo, respecto al valor total por precios, no representa sino el 3% del volumen total. Cundinamarca y Valle del cauca lideran los precios con el 43% y 23% respectivamente del volumen total.

```
departamento_valor = sns.boxplot(x=df["Departamento"],
y=df["Rango_Precio"])
plt.xticks(rotation=90, ha='center')
plt.title("Relación entre Departamento y Rango de Precio",
fontsize=14)
plt.show()
```

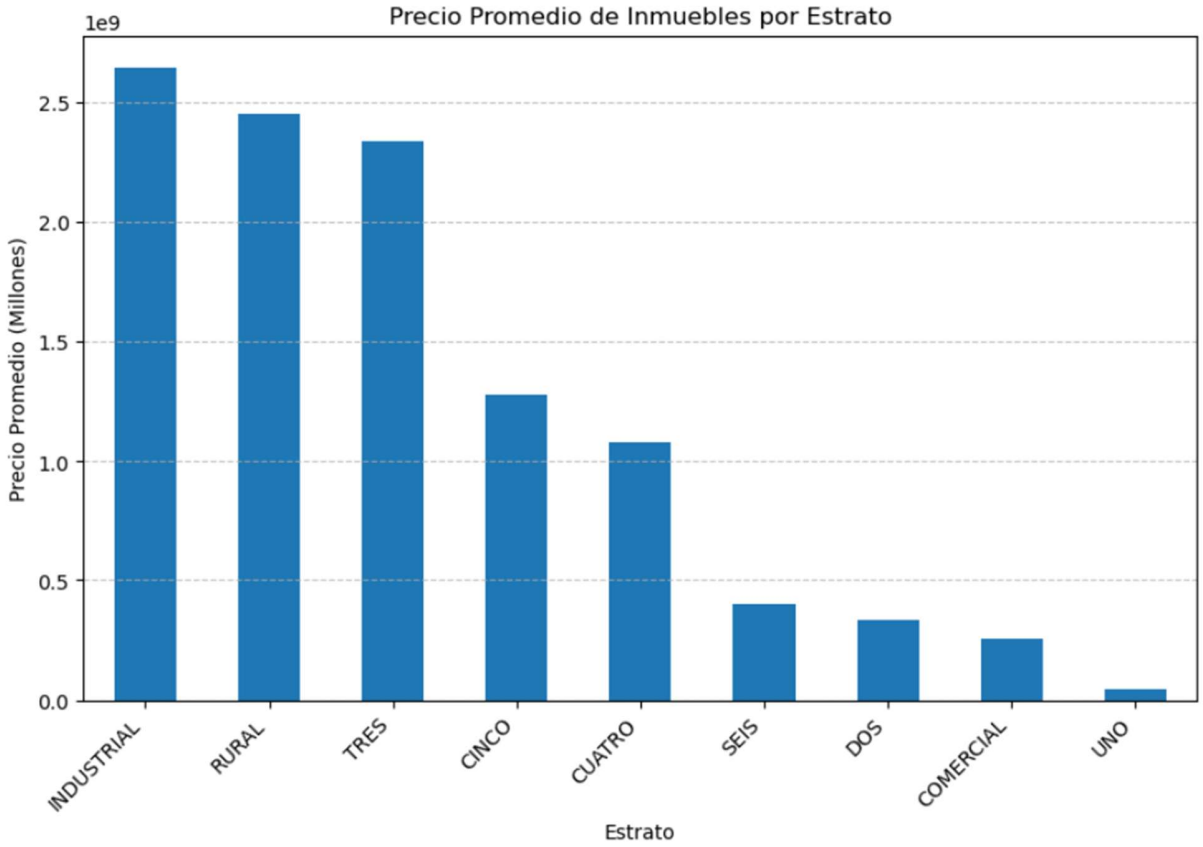


El grafico muestra al departamento de Santander y Bolívar con valores de mediana ligeramente superiores a los demás, sin embargo, como vimos en el grafico anterior estos no concentran más del 6 % del volumen total.

3.¿Qué diferencias existen en los precios promedio de los inmuebles según el estrato socioeconómico declarado?

```
precio_promedio_por_estrato =
df.groupby('Estrato')['Precio'].mean().sort_values(ascending=Fal
se)
plt.figure(figsize=(10, 6))
precio_promedio_por_estrato.plot(
    kind='bar',
```

```
    title='Precio Promedio de Inmuebles por Estrato',
    xlabel='Estrato',
    ylabel='Precio Promedio (Millones)'
)
plt.xticks(rotation=45, ha='right')
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.show()
```



Con las siguientes líneas de codigo podemos calcular estadísticas descriptivas por estrato

```
resumen_estratos = df.groupby('Estrato')['Precio'].agg(['mean',
'median', 'count', 'std'])
resumen_estratos.columns = ['Precio_Promedio', 'Mediana',
'Cantidad_Inmuebles', 'Desviación_Estándar']
print(resumen_estratos.sort_values('Precio_Promedio',
ascending=False))
```

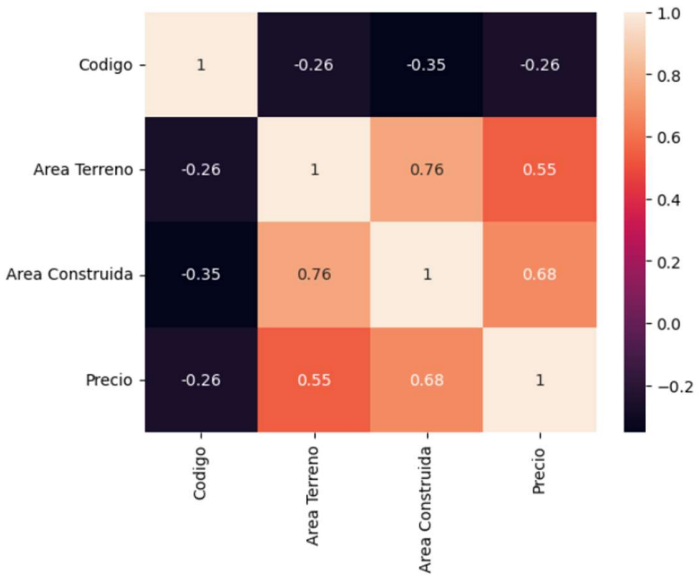
Lo cual nos arroja la siguiente tabla, nótese que los datos de precios y media están en MCOP.

Tabla 6. Estadística descriptiva por estrato

Estrato	Precio_Promedio	Mediana	Cantidad_Inmuebles	Desviación_Estandar
Industrial	2646.41	1196.7	16	6088.79
Rural	2450.88	600.8	40	7213.61
Tres	2335.57	453.1	19	6704.81
Cinco	1278.92	402.2	10	1729.94
Cuatro	1082.89	130.1	19	3125.76
Seis	401.45	213.2	15	522.81
Dos	336.40	99.7	16	497.37
Comercial	255.57	12.3	307	1835.72
Uno	44.38	11.2	21	76.13

Correlación de variables

```
correlacion = df.corr(numeric_only=True)
sns.heatmap(correlacion,xticklabels=correlacion.columns,yticklabels=correlacion.columns,annot=True)
```



Como se observa en el mapa de calor, el área del terreno tiene una correlación del 76% con el área construida, y el precio esta relacionado con ambas en un 55% y 68% respectivamente.

Conclusiones

Partiendo de las preguntas planteadas y los resultados obtenidos en el análisis, se pueden generar las siguientes conclusiones:

- **Los inmuebles comerciales e industriales registran los precios más elevados** debido a su vinculación directa con actividades económicas clave (retail, servicios y producción), lo que justifica su mayor valoración en el mercado.
- **Los estratos con medianas de precio más altas son Rural, Industrial y Cinco**, aunque se detectaron valores atípicos en los estratos Comercial y Seis, lo que sugiere la existencia de propiedades con precios excepcionales fuera de los rangos habituales.
- **El departamento del Meta concentra la mayor cantidad de inmuebles disponibles (más de 250)**, pero solo representa el 3% del valor total del mercado, lo que indica una oferta abundante pero de menor valor promedio en comparación con otras regiones.
- **Cundinamarca y Valle del Cauca dominan en valor económico**, sumando el 66% del volumen total (43% y 23%, respectivamente), lo que refleja su importancia como núcleos de alto valor inmobiliario, a pesar de no tener la mayor cantidad de propiedades.
- **Existe una correlación significativa entre el área del terreno, el área construida y el precio**:
- El área construida está fuertemente relacionada con el área del terreno (76%).
- El precio muestra una dependencia moderada-alta con ambas variables (55% y 68%), lo que resalta la relevancia de estos factores en la valoración de los inmuebles.
- **Nota adicional**: Aunque Santander y Bolívar presentan medianas de precio ligeramente superiores, su participación en el volumen total es mínima (≤6%), lo que sugiere mercados más pequeños pero con propiedades de alto valor puntual.